A Novel Security Protocol for Resolving Addresses in the Location/ID Split architecture

Mahdi Aiash

School of Science and Technology, Middlesex University,
London, UK
{M.Aiash@mdx.ac.uk}

Abstract. The Locator/ID Separation Protocol (LISP) is a routing architecture that provides new semantics for IP addressing. In order to simplify routing operations and improve scalability in future Internet, the LISP uses two different numbering spaces to separate the device identifier from its location. In other words, the LISP separates the 'where' and the 'who' in networking and uses a mapping system to couple the location and identifier. This paper analyses the security and functionality of the LISP mapping procedure using a formal methods approach based on Casper/FDR tool. The analysis points out several security issues in the protocol such as the lack of data confidentiality and mutual authentication. The paper addresses these issues and proposes changes that are compatible with the implementation of the LISP.

Keywords: Location/ID Split Protocol, Casper/FDR, Future Internet, Address Resolving

1 Introduction

Since the public Internet first became part of the global infrastructure, its dramatic growth has created a number of scaling challenges. Among the most fundamental of these is helping to ensure that the routing and addressing systems continue to function efficiently as the number of connected devices increases. To deal with these issues, a number of proposals have been described in the literature such as the LINA, ILNP [1] [2] and the addressing scheme proposed by Aiash et al in [3] [4]. Unlike IP addresses, which combines hosts' locations and identifiers in a single numbering space, the proposals adopted the concept of ID/Location split with uses two separate numbering spaces; one specifies the host's identifier while the other defines its location.

An IETF working group along with the research group at Cisco, are working on the Locator/ID Separation Protocol (LISP) [10]. This protocol shows a great potential; firstly, in addition to dealing with addressing and routing issues, it considers issues like security, QoS, multi-casting and mobility in different environments such as cloud computing and Next Generation Networks (NGNs) [5]. Secondly, large amount of research papers and Internet drafts have been produced by Cisco and the LISP working group which describe the progress in the

design of the LISP [6]. Thirdly, some of the routing and addressing concepts of the LISP have already been implemented in the new Cisco Nexus 7000 Series Switches. Due to these reasons, this paper considers the LISP protocol as an example of the new routing/addressing schemes for future Internet and investigates the security of this protocol.

A key concept of the LISP is that end-systems (hosts) operate the same way they do today. The IP addresses that hosts use for sending and receiving packets do not change. In LISP terminology, these addresses are called Endpoint Identifiers (EIDs). Routers continue to forward packets based on IP destination addresses, the IP addresses of gateway routers or LISP-capable routers at the edge of end-sites are referred to as Routing Locators (RLOCs). To map hosts' EIDs to the authoritative RLOC, the LISP assumes the existence of a mapping or address resolving system that consists of a Map Server (MS) and a distributed database to store and propagate those mappings globally. The functionality of the mapping system goes through two stages:

1. Registration Stage: in this stage, the Map Server learns the EIDs-to-RLOC mappings from an authoritative LISP-Capable Router and publishes them in the database.
2. Addresses resolving Stage: the Map Server (Ms) accepts Map-Requests from routers, looks up the database and returns the requested mapping.

These two stages will be explained in more details in section 2.2.

Currently, the research concentrates mainly on defining the LISP architecture as well as the structure of the packets such as the Map-Request and Map-Reply messages. However, the security-related research is still at an early stage, the research in [7] [8] have highlighted potential threats as an introduction to come up with the required security mechanisms. These research efforts have not defined specific attacks against the deployment of the LISP. Therefore, this paper uses formal methods approach based on the well developed CASPER/FDR [15] tool to investigate the security of implementing the LISP architecture. Our main concern here is the security of the address resolving stage (stage 2), where a LISP-capable router approaches the Map Server with a Map-Request message and expects the required EID-to-RLOC mapping in a Map-Replay message.

This study adds the following contributions: firstly, using formal methods approach, it discovers and describes possible attacks against the implementation of the LISP architecture. Secondly, to fix these problems, the paper proposes feasible solution that is in line with the goals of the LISP's security requirements as defined in [8]. The proposed solution has been formally verified using Casper/FDR. We believe that, this paper will help researchers and developers to realize some of the actual security threats and use the proposed solution as a guideline to come up with the most complete security solutions.

The rest of the paper is organised as follows: Section 2 describes related work in the literature. Section 3 formally analyses the security of the basic address procedure of the LISP, then using a progressive approach, it explains and formally verifies the refinement stages, which led to the final version of the secure protocol. The paper is concludes in Section 4.

2 Related Work

2.1 An Overview of The LISP

To improve routing scalability while facilitating flexible address assignment in multi-homing and mobility scenarios, the LISP describes changes to the Internet architecture in which IP addresses are replaced by routing locators (RLOCs) for routing through the global Internet and by endpoint identifiers (EIDs) for identifying network sessions between devices [9]. As shown in Fig 1, three essential components exist in the LISP environment: the LISP sites (EID space), the non-LISP sites (RLOC space), and the LISP Mapping System which comprises Map Servers and databases.

- **The LISP sites (EID space):** they represent customer end-sites in exactly the same way that end-sites are defined today. However, the IP address in the EID space are not advertised to the non-LISP sites, but are published into the LISP Mapping Systems which performs the EID-to-RLOC mapping. The LISP functionalities is deployed on the site's gateway or edge routers. Therefore, based on their roles, two types of routers are defined: firstly, the Ingress Tunnel Routers (ITRs) which receive packets from hosts and send LISP packets toward the Map Server. Secondly, the Egress Tunnel Routers (ETRs) which receive LISP packets from the Map Server and pass them to hosts [10] [9].
- **Non-LISP sites (RLOC space):** it represents current sites where the IP addresses are advertised and used for routing purpose.
- **LISP Mapping Systems:** These are represented by Map Servers (MS) and a globally distributed database that contains all known EID prefixes to RLOC mappings. Similar to the current Domain Name System (DNS), the Mapping systems are queried by LISP-capable devices for EID-to-RLOC mapping.

2.2 Interactions With Other LISP Components

The functionality of the LISP goes through two stages:

1. **The EID Prefix Configuration and ETR Registration Satge:**
   As explained in [11], an ETR publishes its EID-prefixes on a Map Server (MS) by sending LISP Map-Register messages which includes the ETR's RLOC and a list of its EID-prefixes. Initially, it has been presumed that prior to sending a Map-Register message, the ETR and Map Server must be configured with a shared secret or other relevant authentication information. Upon the receipt of a Map-Register from an ETR, the Map Server checks the validity of the Map-Register message and acknowledges it by sending a Map-Notify message. When registering with a Map-Server, an ETR might request a no-proxy reply service which implies that the Map Server will forward all the EID-to-RLOC mapping requests to the relevant ETR rather than dealing with them.

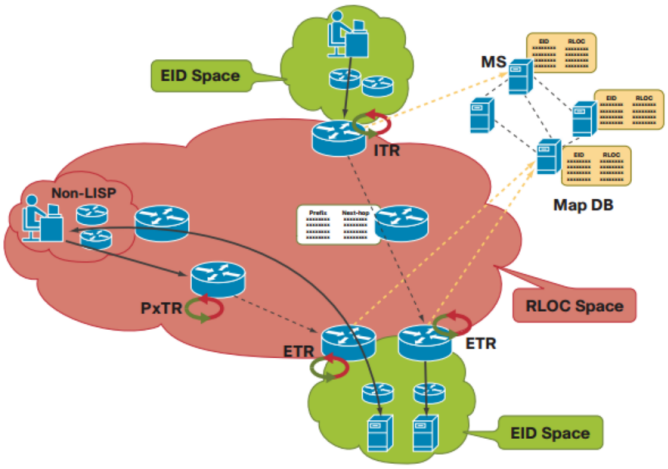

Fig. 1. The LISP Network Architecture Design [9]

The registration stage, shown in Fig 2, is vulnerable to serious security threats such as replay and routing table poisoning attacks. A detailed security analysis of this stage has been presented in another work of our group in [12].

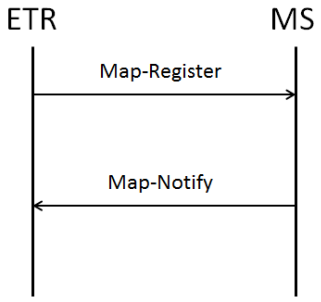

Fig. 2. The ETR Registration Process

2. **The Address Resolving Stage:** Once a Map Server has EID-prefixes registered by its client ETRs, it will accept and process Map-Requests. In response to a Map-Request (sent from an ITR), the Map Server first checks to see if the required EID matches a configured EID-prefix. If there is no match, the Map Server returns a negative Map-Reply message to the ITR. In case of a match, the Map Server re-encapsulates and forwards the resulting Encapsulated Map-Request to one of the registered ETRs which will return Map-Replay directly to the requesting ITR as shown in Fig 3.

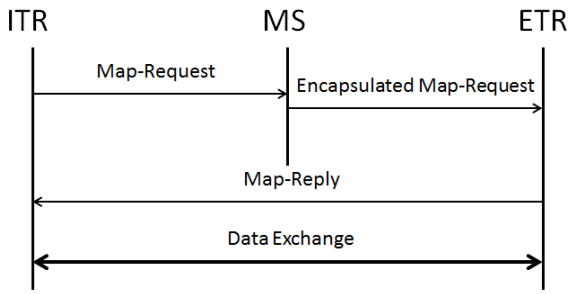

Fig. 3. The No Proxy Map Server Processing

The LISP working group in [10] has defined the structure of all the LISP Packets including the Map-Request, the Map-Notify, the Map-Register and the MAP-Reply. However, for the security analysis in section 3, only security-related parameters of the LISP messages are explicitly mentioned.

2.3 Verifying Security Protocols using Casper/FDR

Previously, analysing security protocols used to be done using two stages. Firstly, modelling the protocol using a theoretical notation or language such as the CSP [13]. Secondly, verifying the protocol using a model checker such as Failures-Divergence Refinement (FDR) [14]. However, describing a system or a protocol using CSP is a quite difficult and error-prone task; therefore, Gavin Lowe [15] has developed the CASPER/FDR tool to model security protocols, it accepts a simple and human-friendly input file that describes the system and compiles it into CSP code which is then checked using the FDR model checker. Casper/FDR has been used to model communication and security protocols as in [16], [17]. The CASPER's input file that describes the systems consists of eight headers as explained in Table 1.

3 Analysing the Security of the Address Resolving Procedure

3.1 System Definition

As shown in Fig 3, and based on the notations in Table 2, the procedure of the mapping procedure goes as follows:

Msg1. ITR→ MS : ITR, N1, MapRequest, h(ITR, N1, MapRequest)

The ITR sends a Map-Request message which includes a 4-byte random nonce (N1) and the addresses of the ITR. The ITR expects to receive the same nonce in the Map-Reply message.

Table 1. THE HEADERS OF CASPER'S INPUT FILE

The Header	Description
# Free Variables	Defines the agents, variables and functions in the protocol
# Processes	Represents each agent as a process
# Protocol Description	Shows all the messages exchanged between the agents
# Specification	Specifies the security properties to be checked
# Actual Variables	Defines the real variables, in the actual system to be checked
# Functions	Defines all the functions used in the protocol
# System	Lists the agents participating in the actual system with their parameters instantiated
# Intruder Information	Specifies the intruder's knowledge and capabilities

Msg2. MS→ ETR : ITR, N1, MapRequest, h(ITR, N1, MapRequest)

The Map Server (MS) encapsulates Msg1 and passes it to the relevant ETR as Msg2.

Msg3. ETR→ ITR : ETR, N1, MapReply, h(ETR, N1, MapReply)

The ETR composes Msg3 which includes a Map-Reply and the received nonce (N1). Upon receiving this message, the ITR checks the included nonce and only when the check succeeds, the ITR authenticates the ETR.

Table 2. Notation

The Notation	Definition
ITR	The Ingress Tunnel Router in the source EID Space
ETR	The Egress Tunnel Router in the destination EID Space
MS	The Map Server
N1	The Nonce
h(m)	Hash value of the message (m)
{m}{K}	The message (m) being encrypted with the key (K)

3.2 Formal Analysis of the Basic Mapping Procedure

To formally analyse the basic mapping procedure, we simulate the system using Casper/FDR tool. A Casper input file describing the system in Figure 3 was prepared. for conciseness, only the #Specification and the #Intruder Information headings are described here, while the rest are of a less significance in terms of understanding the verification process.

The security requirements of the system are defined under the # Specification heading. The lines starting with the keyword **Secret** define the secrecy properties of the protocol. The `Secret(ITR, N1, [Ms, ETR])` specifies the N1 nonce as a secret between ITR, Ms and ETR. The lines starting with **Agreement** define the protocol's authenticity properties; for instance `Agreement(ETR, ITR, [N1])` specifies that, the ETR is correctly authenticated to ITR using the random number N1. The `WeakAgreement(ITR, Ms)` assertion could be interpreted as follows: if ITR has completed a run of the protocol with Ms, then Ms has previously been running the protocol, apparently with ITR.

```
#Specification
Secret(ITR, N1, [Ms, ETR])
WeakAgreement(ITR, Ms)
WeakAgreement(ITR, ETR)
WeakAgreement(ETR, ITR)
Agreement(ETR, ITR, [N1])
```

The # Intruder Information heading specifies the intruder identity, knowledge and capability. The first line identifies the intruder as Mallory, the intruder knowledge defines the Intruder's initial knowledge, i.e., we assume the intruder knows the identity of the participants and can fabricate Map Request and Map Reply messages.

```
#Intruder Information
Intruder = Mallory
IntruderKnowledge = {ITR, ETR, Ms, Mallory, mapRequest, mapReply}
```

After generating the CSP description of the systems using Casper and asking FDR to check the security assertions. The following attacks were found:

1. The First attack is against the `WeakAgreement(ITR, Ms)` assertion, and it goes as follows:
   ```
   1. ITR -> I_Ms : ITR, N1, mapRequest, h(ITR, N1, mapRequest)
   1. I_ETR -> Ms : ETR, N1, mapReply, h(ETR, N1, mapReply)
   2. Ms -> I_ETR : ETR, N1, mapReply, h(ETR, N1, mapReply)
   ```

   Where the notations I_ Ms, I_ETR and I_ITR represent the case where the Intruder impersonates the Ms, ETR and ITR, respectively. This is an active Man-in-the-Middle attack; the Intruder blocks the first message and composes message two, acting as the ETR. Upon receiving this message, the Map Sever mistakenly believes that the message came from ETR and hence replies with a Map-Replay message, which will be intercepted by the Intruder.
2. The second attack compromises three assertions `Secret(ITR, N1, [Ms, ETR])`, `Agreement(ETR, ITR, [N1])`, `WeakAgreement(ETR, ITR)`, and it goes as follows:

```
1. ITR -> I_Ms : ITR, N1, MapRequest, h(ITR, N1, MapRequest)
3. I_ETR -> ITR : ETR, N1, MapReply, h(ETR, N1, MapReply)
The intruder knows N1
```

In this attack, the intruder intercepts the first message and replays to the ITR acting as ETR. Since there is no encryption, the Intruder acquires the nonce N1 and uses it to impersonate ETR; consequently, the ITR runs this process believing it is with ETR while in reality it is with the Intruder. Furthermore, the basic protocol uses the nonce N1 to authenticate the ETR to the ITR. However, it does not provide any approach to authenticate the ITR to the ETR.

The discovered attacks are due to the lack of security in the transaction between the participating parties. Therefore, the following subsections will propose security measures to address the discovered attacks.

3.3 The First Proposed Enhancement

The first discovered attack in section 3.2 was due to the exposure of the nonce (N1). Therefore, to stop this attack, there is a need to secure the (ITR-MS) and the (MS-ETR) connections. As explained in section 2, for the Registration process, it is presumed that LISP-Capable routers (ITR, ETR) and MS have already agreed on secret keys. Similarly, we will presume that these keys will be used to secure the transactions in the resolving procedure. Hence, two pre-configured secret keys: (K1) is shared between ITR and MS, and (K2) is shared between the MS and ETR. The enhanced version of the protocol looks as follows:

Msg1. ITR→ MS : {ITR, N1, MapRequest, h(ITR, N1, MapRequest)}{K1}
Msg2. MS→ ETR : {ITR, N1, MapRequest, h(ITR, N1, MapRequest)}{K2}
Msg3. ETR→ ITR : ETR, N1, MapReply, h(ETR, N1, MapReply)

We modelled the new version of the protocol with Casper and checked it with FDR, the following attack against the secrecy assertion was discovered.

```
1a. ITR -> I_Ms : {ITR, N1, mapRequest, h(ITR, N1, mapRequest)}{K1}
1b. I_ITR -> Ms : {ITR, N1, mapRequest, h(ITR, N1, mapRequest)}{K1}
2a. Ms -> I_ETR : {ITR, N1, mapRequest, h(ITR, N1, mapRequest)}{K2}
2b. I_Ms -> ETR : {ITR, N1, mapRequest, h(ITR, N1, mapRequest)}{K2}
3a. ETR -> I_ITR : ETR, N1, mapReply, h(ETR, N1, mapReply)
3b. I_ETR -> ITR : ETR, N1, mapReply, h(ETR, N1, mapReply)
The intruder knows N1
```

Here, the Intruder passively replays the messages between the participants. This attack could be interpreted as follows: the ITR will complete running the protocol believing that it was with the ETR, while it was with the Intruder instead. Similarly, the ETR will believe it has been running the protocol with the ITR,

while in reality it was with the Intruder. Again, this attack is ascribed to the exposure of the nonce (N1), which highlight the need for securing the direct transaction between the ITR and ETR. Also, there is a need to propose an authentication mechanism, through which the ETR can authenticate the ITR.

3.4 The Final Enhancement: The Proposed AKA Protocol

In order to secure the direct connection between the ITR and ETR, and to achieve a mutual authentication between them. We propose an Authentication and Key Agreement (AKA) protocol that does not require major modifications to the basic LISP protocol. The proposed AKA protocol is based on the Challenge-Response paradigm and it goes as follows:

Msg1. ITR→ MS:{ITR,N1,MapRequest,K3,h(ITR, N1,MapRequest,K3)}{K1}
Msg2. MS→ ETR:{ITR,N1,MapRequest,K3, h(ITR,N1,MapRequest,K3)}{K2}

The ITR composes Msg1 and includes a freshly generated secret key (K3) to be used by the ETR to encrypt the Map-Reply packet. This message is forwarded by the MS towards the ETR.

Msg3. ETR→ ITR:{ETR, N1, N2 MapReply, h(ETR, N1,N2 MapReply)}{K3}

Upon receiving the Map-Request in Msg2, the ETR replies with a Map-Reply message with a challenge nonce (N2). The message is encrypted using the suggested key (K3).

Msg4. ITR → ETR : {N2}{K3}

The ITR returns the challenge (N2) encrypted using the key (K3). The ETR will check the returned challenge to authenticate ITR.

To verify the proposed AKA protocol, we prepared a Casper file that describes the protocol (the full Casper input file is shown in the Appendix). To check the mutual authentication, the `Agreement(ITR, ETR, [N2])` assertion has been added to the # Specification heading as shown below:

```
#Specification
Secret(ITR, N1, [Ms, ETR])
WeakAgreement(ITR, Ms)
WeakAgreement(ITR, ETR)
WeakAgreement(ETR, ITR)
Agreement(ETR, ITR, [N1])
Agreement(ITR, ETR, [N2])
```

We simulated this security considerations with Casper and asked FDR to check for attacks. Casper/FDR failed to find attacks against any of the checked assertions as shown in Fig 4.

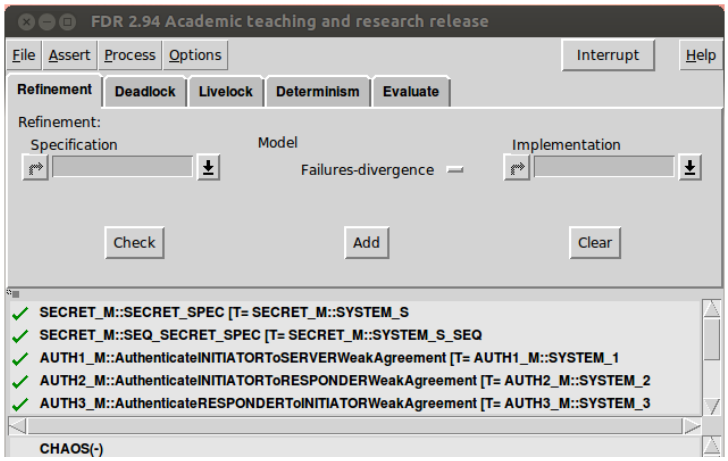

Fig. 4. The FDR Verification

Protocol Analysis: The main goals of the proposed protocol are to achieve mutual authentication between ETR and ITR and to secure the direct connection between them. Furthermore, it is crucial to achieve these goals with a minimum modification to the basic LISP. The security-related goals could be achieved using different protocols, examples of there are the Internet Key Exchange (IEK) [18], and Virtual Private Network (VPN) protocols such as the Internet Protocol Security (IPsec) [19]. However, these protocols will increase the number of exchanged messages significantly, At least five extra messages in the case of IKE and more than this in the case of IPSec (based on the IPSec mode). Furthermore, packets-encapsulation due to the tunnelling process in VPN protocols will lead to adding extra headers to the LISP packets which make them incompatible with the current implementation of the LISP-capable devices.

The fact that the formal verification of the proposed protocol, using Casper/FDR, found no attacks against any of the checked assertions, implies that the protocol successfully achieves a number of crucial security requirements such as mutual authenticating the participating parties and maintaining the secrecy of the session key between the ITR and ETR. Furthermore, the protocol does not require major modification to the basic LISP transactions and no extra headers are needed for packets encapsulation.

4 Conclusion

This paper analysed the security of the address resolving process in LISP protocol. Analysing and verifying the basic LISP using Casper/FDR shows that the protocol is vulnerable to authentication and secrecy attacks. Therefore, a new security protocol was introduced in this article, the article described the refinement stages of the protocol along with the discovered attacks. The final version

of the proposed protocol was proven to be secure and to comply with the design of the LISP protocol.

References

1. Ishiyama, I., Uehara, K., Esaki, H., Teraoka, F.: LINA: ANew Approach to Mobility in Wide Area Networks. In IEICE Trans. Commun. , vol. E84-B, no. 8, August 2001.
2. Atkinson, RJ.:ILNP Concept of Operations. 27 July 2011, internet Draft.
3. Mapp, G., Aiash, M., Crestana Guardia, H., Crowcroft, J.: Exploring Multi-homing Issues in Heterogeneous Environments 1st International Workshop on Protocols and Applications with Multi-Homing Support(PAMS2011) Singapore (2010).
4. Aiash, M., Mapp, G., Lasebae, A., Phan, R., Augusto, M., Vanni, R., Moreira,E.: Enhancing Naming and Location Services to supportMulti-homed Devices in Heterogeneous Environments. In Proc. TheCCSIE 2011, London-UK, 25-27 July (2011).
5. Cisco Nexus 7000 Series Switches, Cisco Nexus 7000 LISP Overview Video: `http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6554/ps6599/ps10800/LISP_VDS.html`. [Last Accessed on 13.01.13].
6. Locator/ID Separation Protocol (lisp) Working Group. `http://datatracker.ietf.org/wg/lisp/charter/`, [Last Accessed on 13.01.13].
7. Cisco Locator/ID Separation Protocol Security At-A-Glance.`http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6554/ps6599/ps10800/at_a_glance_c45-645204.pdf`. [Last Accessed on 13.01.13].
8. Maino, F., Ermagan, V., Cabellos, A., Saucez, A., Bonaventure, O.: LISP-Security (LISP-SEC). Internet-Draft , September 12, 2012.
9. Cisco Locator/ID Separation Protocol Revolutionary Network Architecture to Power the Network. `http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6554/ps6599/ps10800/aag_c45-635298.pdf`. [Last Accessed on 13.01.13].
10. Farinacci, D., Fuller, V., Meyer, D., Lewis, D.: Locator/ID Separation Protocol (LISP). Internet-Draft , November 13, 2012.
11. Farinacci, D., Fuller, V.: LISP Map Server Interface. Internet-Draft , March 4, 2012.
12. Aiash, M.: Securing Address Registration in Location/ID Split Protocol using ID-Based Cryptography. Submitted to The 11th International Conference on Wired/Wireless Internet Communications WWIC 2013.
13. Goldsmith, M., Lowe, G., Roscoe, A.W., Ryan,P., Schneider, S.: The modelling and analysis of security protocols, PEARSON Ltd, 2010.
14. Formal Systems, Failures-divergence refinement. fdr2 user manual and tutorial, June 1993, Version 1.3.
15. Lowe, G., Broadfoot, P., Dilloway, C., Hui, M. L.: Casper: A compiler for the analysis of security protocols, 1.12 ed., September 2009.
16. Aiash, M., Mapp, G., Lasebae, A., Phan, P., Loo, J.: Casper: A formally verified AKA protocol for vertical handover in heterogeneous environments using Casper/FDR, EURASIP Journal on Wireless Communications and Networking 2012, 2012:57.
17. Aiash, M., Mapp, G., Lasebae, A., Phan, P., Loo, J.: A Formally Verified Device Authentication Protocol Using Casper/FDR. In 11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 25-27 June 2012.

18. Harkins, D., Carrel, D.: The Internet Key Exchange (IKE). Request for Comments: 2409, November 1998.
19. Kent, S., Atkinson, R.: IP Encapsulating Security Payload (ESP). Request for Comments: 2406, November 1998.

Appendix: The Final Version of the Protocol

```
#Free variables
Itr, Etr : Agent
na, nb, seq2, n1, n2 : Nonce
K1, K2: PreSharedKey
Ms: Server
K3: SessionKey
MappRequest,MappReply: Messages
InverseKeys = (K3,K3),(K2, K2), (K1, K1)
h : HashFunction
#Processes
INITIATOR(Itr,Ms,Etr,n1, MappRequest, K1, K3)
SERVER(Ms, Etr, K1, K2)
RESPONDER(Etr, MappReply, K2, n2)
#Protocol description
0. -> Ms : Itr
1. Itr -> Ms : {Itr, n1,MappRequest, K3, h(Itr, n1, MappRequest)}{K1}
2. Ms -> Etr : {Itr, n1,MappRequest, K3, h(Itr, n1, MappRequest)}{K2}
3. Etr -> Itr : {Etr, n1,MappReply,n2, h(Etr, n1, MappReply)}{K3}
4. Itr -> Etr : {n2}{K3}
#Specification
Secret(Itr, n1, [Ms, Etr])
WeakAgreement(Itr, Ms)
WeakAgreement(Itr, Etr)
WeakAgreement(Etr, Itr)
Agreement(Etr, Itr, [n1])
Agreement(Itr, Etr, [n2])
#Actual variables
itr, etr, Mallory : Agent
Na, Nb, Seq2, N1, N2 : Nonce
k1, k2: PreSharedKey
ms: Server
mappRequest,mappReply: Messages
InverseKeys = (k2, k2), (k1, k1), (k3,k3)
k3: SessionKey
#System
INITIATOR(itr,ms, etr, N1, mappRequest, k1, k3)
SERVER(ms, etr, k1, k2)
RESPONDER(etr, mappReply, k2, N2)
```

```
#Intruder Information
Intruder = Mallory
IntruderKnowledge = {itr, etr, ms, Mallory, mappRequest, mappReply}
```